

# Header-gen 安装使用说明

安装前所需

LLVM/CLANG 3.3

Make 3.82 及以上版本

Python-termcolor(可以在 header-gen 中 contrib/termcolor-1.1.0 中找到并安装)

Sqlite3-devel

CMake 2.8

sqlite3

sqlite3-devel

## 安装 LLVM/CLANG3.3

### 1. 下载 llvm3.3

```
cd where-you-want-llvm-to-live
```

```
svn co http://llvm.org/svn/llvm-project/llvm/tags/RELEASE_33/final llvm
```

### 2. 下载 CLANG3.3

```
cd where-you-want-llvm-to-live
```

```
cd llvm/tools
```

```
svn co http://llvm.org/svn/llvm-project/cfe /tags/RELEASE_33/final clang
```

### 3. 下载 Compiler-RT

```
cd where-you-want-llvm-to-live
```

```
cd llvm/projects
```

```
svn co http://llvm.org/svn/llvm-project/compiler-rt/tags/RELEASE_33/final compiler-rt
```

### 4. 将 header-gen/patch/clang 的补丁打到 clang 里

### 5. 编译安装 LLVM/CLANG

```
cd where-you-want-to-build-llvm
```

```
mkdir build
```

```
cd build
```

```
../llvm/configure --enable-optimized
```

```
make
```

```
make install
```

!! 注意：如果已安装好的 LLVM3.3/CLANG3.3 仍需将补丁打到 clang 里，重新编译安装

## header-gen 安装

1. 将安装好的 llvm/clang 中的 build/Release+Asserts 下的 bin 和 lib ( or lib64 ) 文件夹加入

到 header-gen 文件夹中,加入后应该文件夹如下：

```
header-gen
├── bin
├── clang-plugins
├── contrib
├── DeclComposer.py
├── envsetup.sh
├── lib
├── linux
├── Makefile.inc
├── patch
├── README
└── unittests
```

2. 编译安装 DeclFilter.so.

```
cd header-gen/clang-plugins
vi CMakeLists.txt
```

将 LLVM\_SRC\_DIR, CLANG\_SRC\_DIR, LLVM\_BUILD\_DIR and CLANG\_BUILD\_DIR

修改为本地的地址，如：

```
set (LLVM_SRC_DIR /home/os/softs/llvm3.3/llvm)
set (CLANG_SRC_DIR /home/os/softs/llvm3.3/llvm/tools/clang)
set (LLVM_BUILD_DIR /home/os/softs/llvm3.3/build/Release+Asserts)
set (CLANG_BUILD_DIR /home/os/softs/llvm3.3/build/tools/clang)
```

3. 编译安装，执行：

```
[xx@xx clang-plugins]$ mkdir build
[xx@xx clang-plugins]$ cd build
[xx@xx build]$ cmake ..
[xx@xx build]$ make
[xx@xx build]$ cp lib/*.so ../..
```

安装完成

## header-gen 使用说明

如要想对 linux 内核中的某一驱动进行使用

1. 修改 envsetup.sh,如下

```
export LINUX_DIR=/home/fluto/softs/linux-kernel/linux3.12 (内核源码位置)
```

```
export ARCH=x86
```

```
export BOARD=
```

```
export TOOLCHAIN_PREFIX=
```

```
export PLATFORM_CC_FLAGS=""
```

2. 执行:

```
[xx@xx header-gen]$ source envsetup.sh (每次重新开启 bash 都需要重新执行)
```

3. 将某一驱动的文件夹放入到 header-gen/linux 中

4. 将 linux 目录下的 Makefile 进行修改,将你所需要进行驱动目录名加入到 directories 中,

如:

```
directories = e1000 ixgbevf virtio bcm2708
```

5. 执行:

```
make e1000.o
```

结束